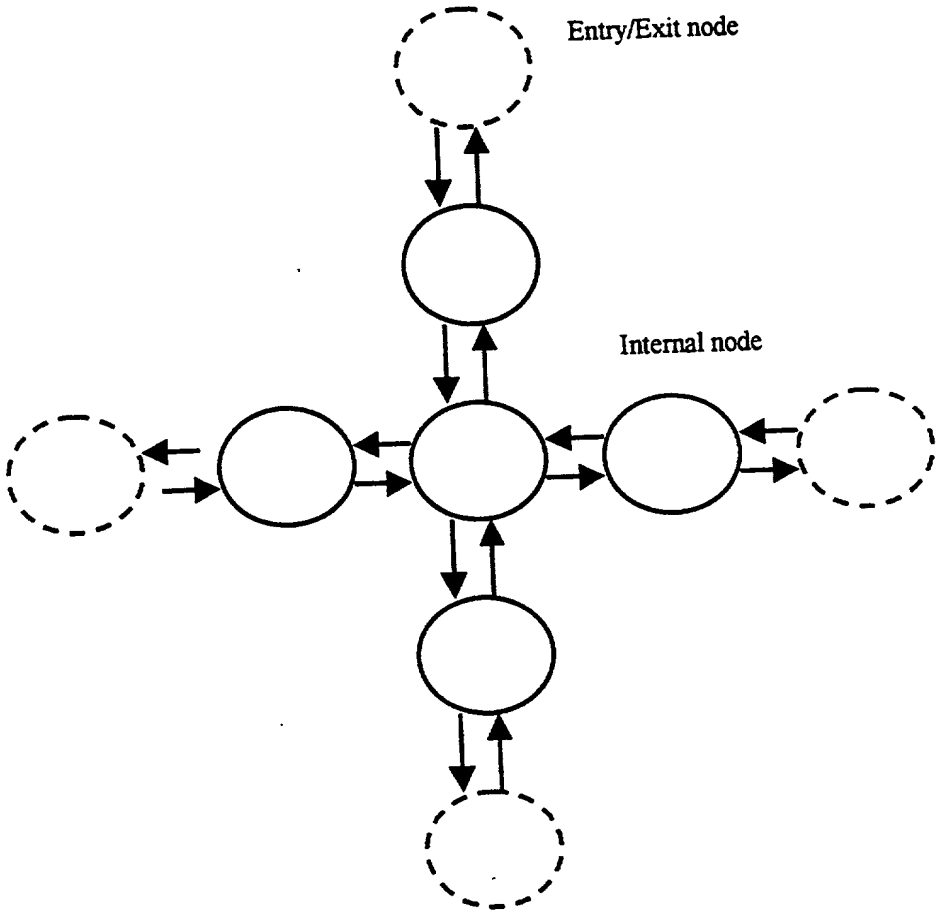


Fig. 1

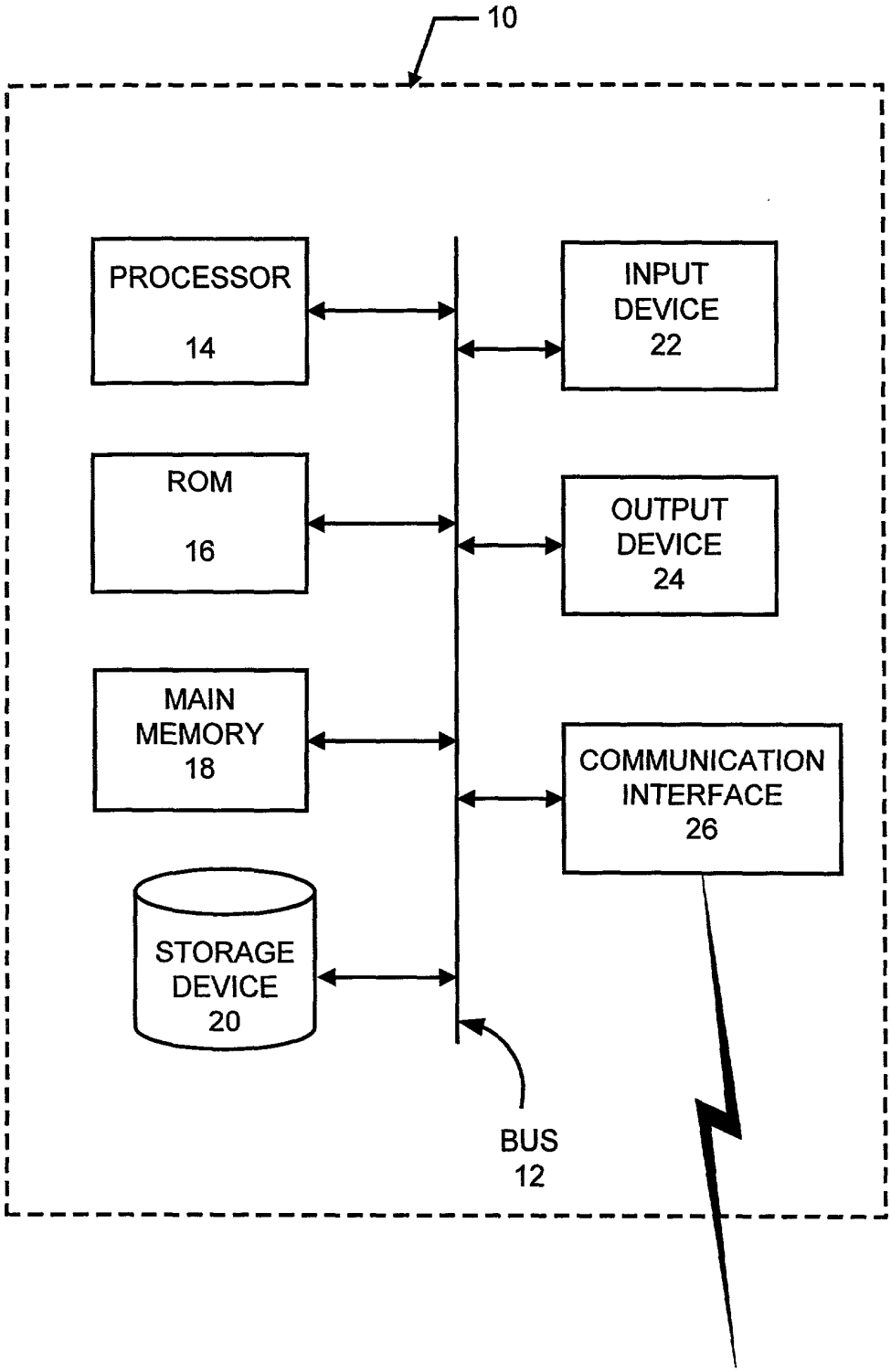

10
PROCESSOR
14
ROM
16
MAIN
MEMORY
18
STORAGE
DEVICE
20
INPUT
DEVICE
22
OUTPUT
DEVICE
24
COMMUNICATION
INTERFACE
26
BUS
12

Fig. 2

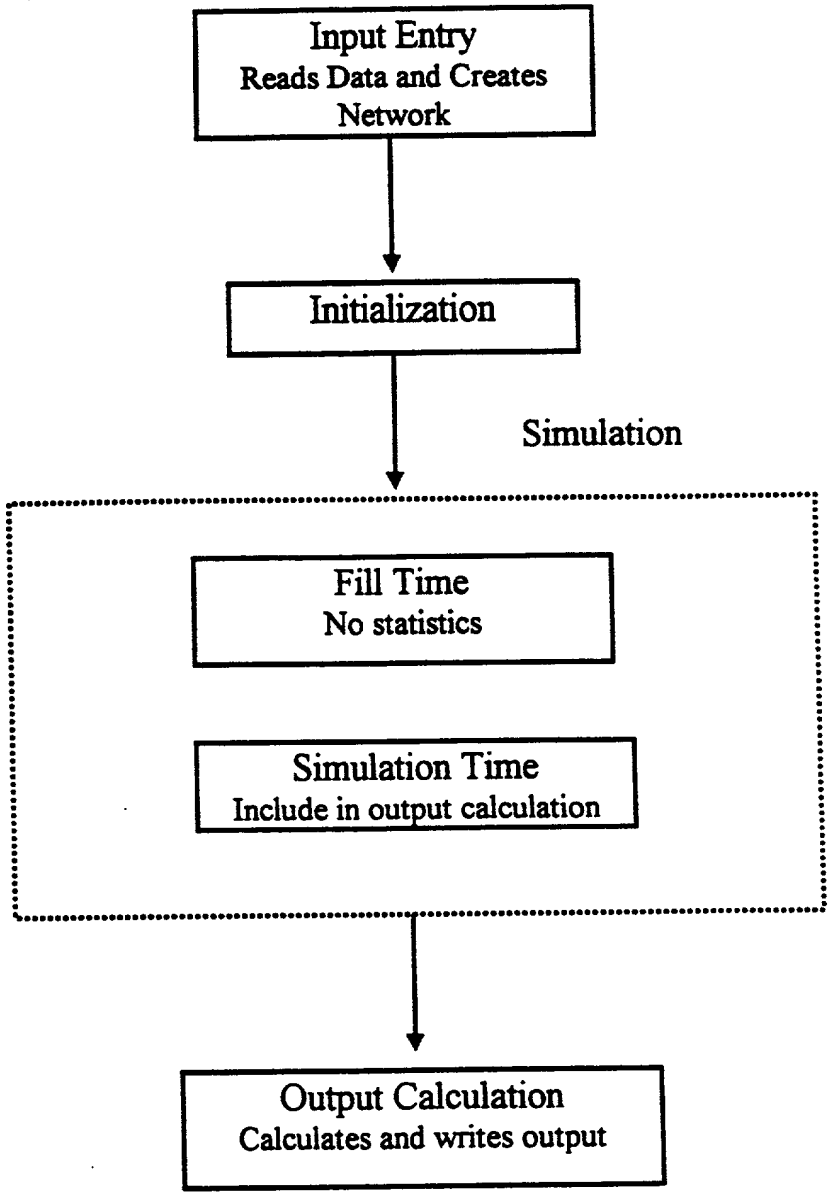

Input Entry
Reads Data and Creates Network
Initialization
Simulation
Fill Time
No statistics
Simulation Time
Include in output calculation
Output Calculation
Calculates and writes output

Fig. 3

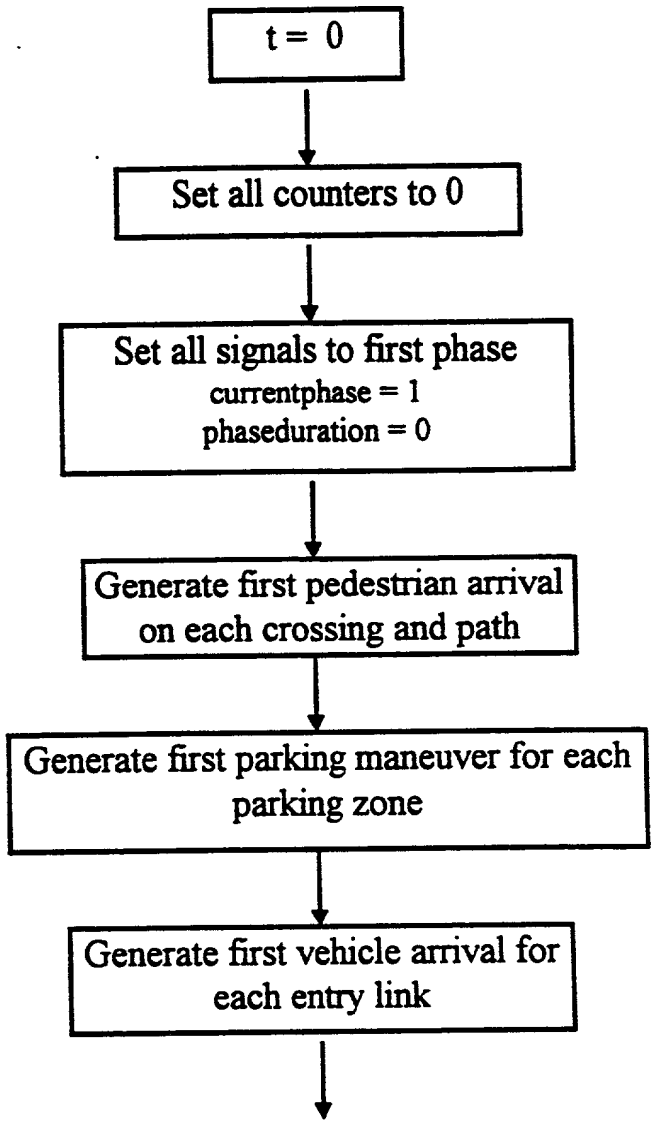

t = 0
Set all counters to 0
Set all signals to first phase
currentphase = 1
phaseduration = 0
Generate first pedestrian arrival on each crossing and path
Generate first parking maneuver for each parking zone
Generate first vehicle arrival for each entry link

Fig. 4

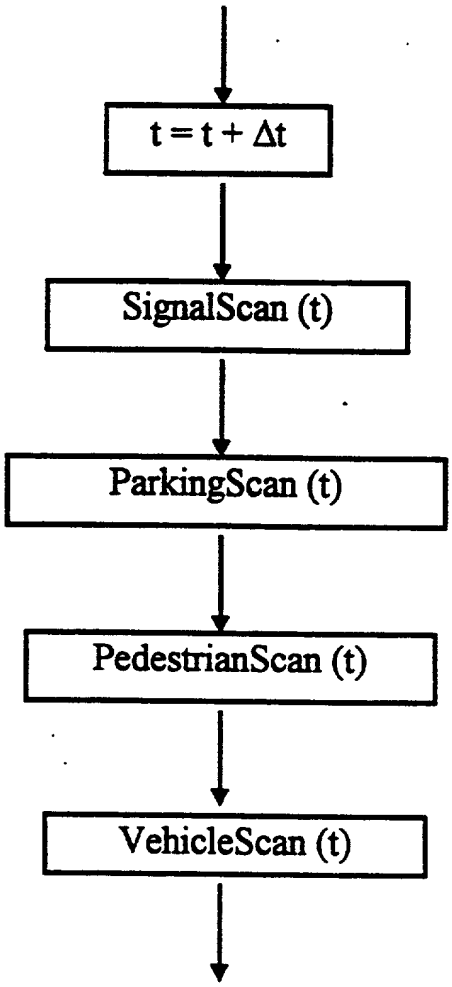

t = t + Δt
SignalScan (t)
ParkingScan (t)
PedestrianScan (t)
VehicleScan (t)

Fig. 5

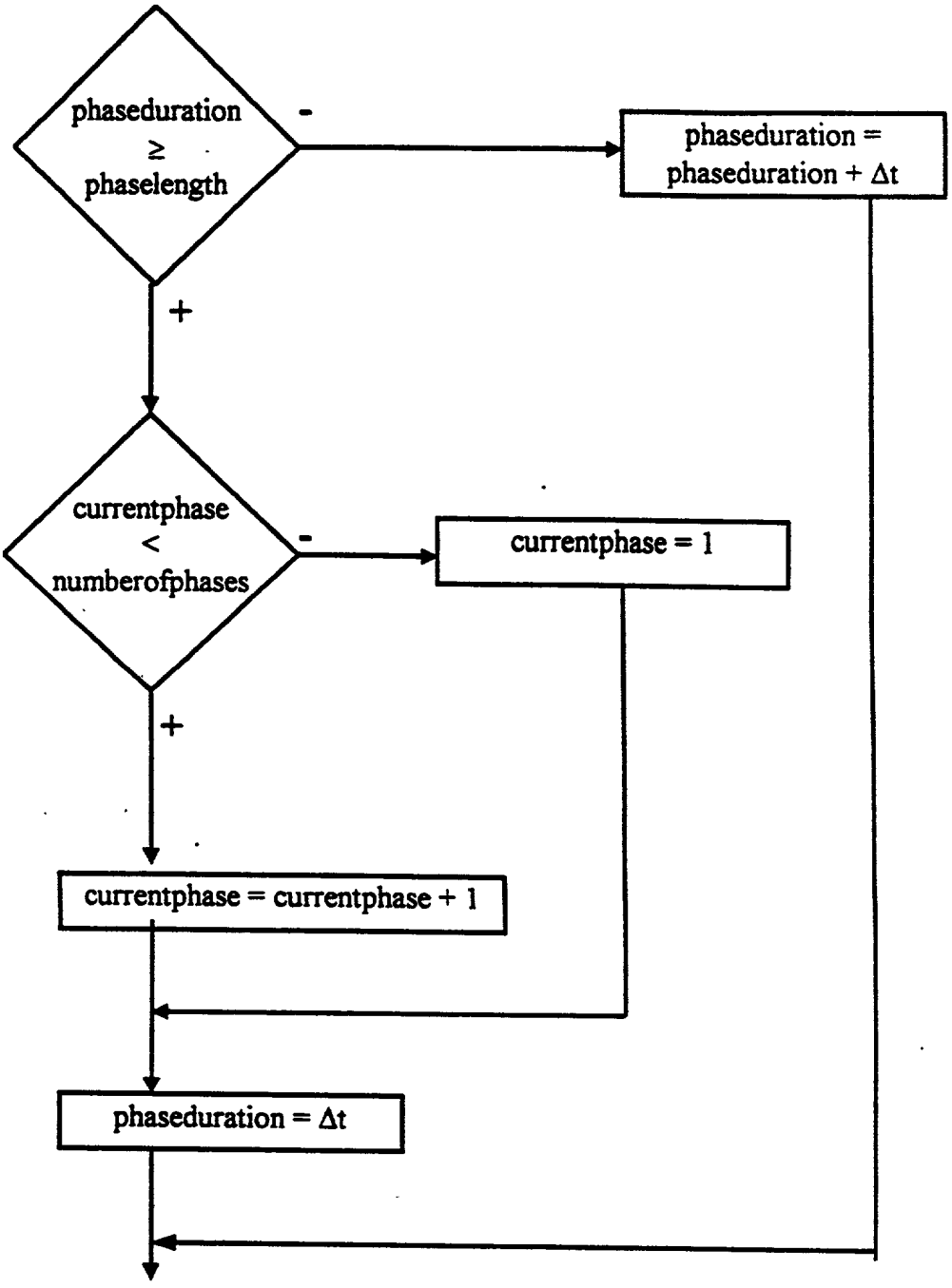

phaseduration ≥ phaselength
-
phaseduration = phaseduration + Δt
+
currentphase < numberofphases
-
currentphase = 1
+
currentphase = currentphase + 1
phaseduration = Δt

Fig. 6

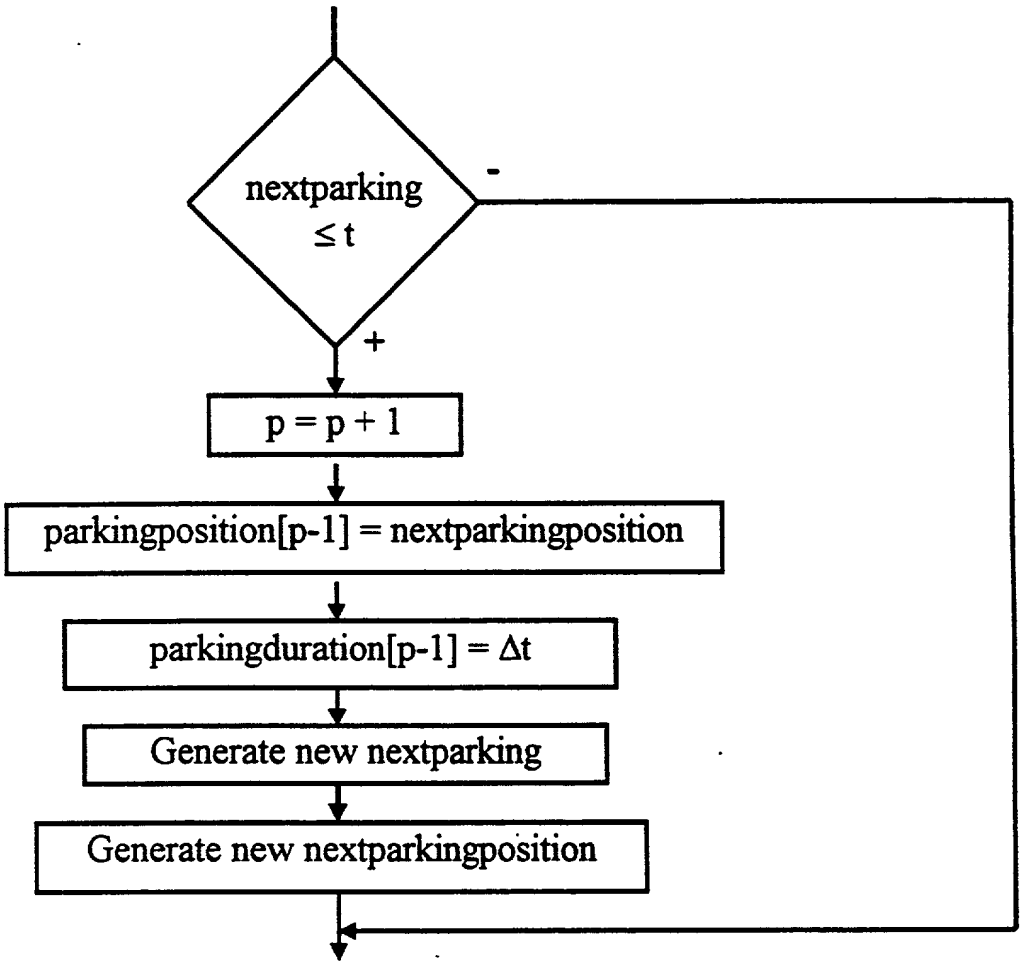

nextparking
≤ t
-
+
p = p + 1
parkingposition[p-1] = nextparkingposition
parkingduration[p-1] = Δt
Generate new nextparking
Generate new nextparkingposition

Fig. 7A

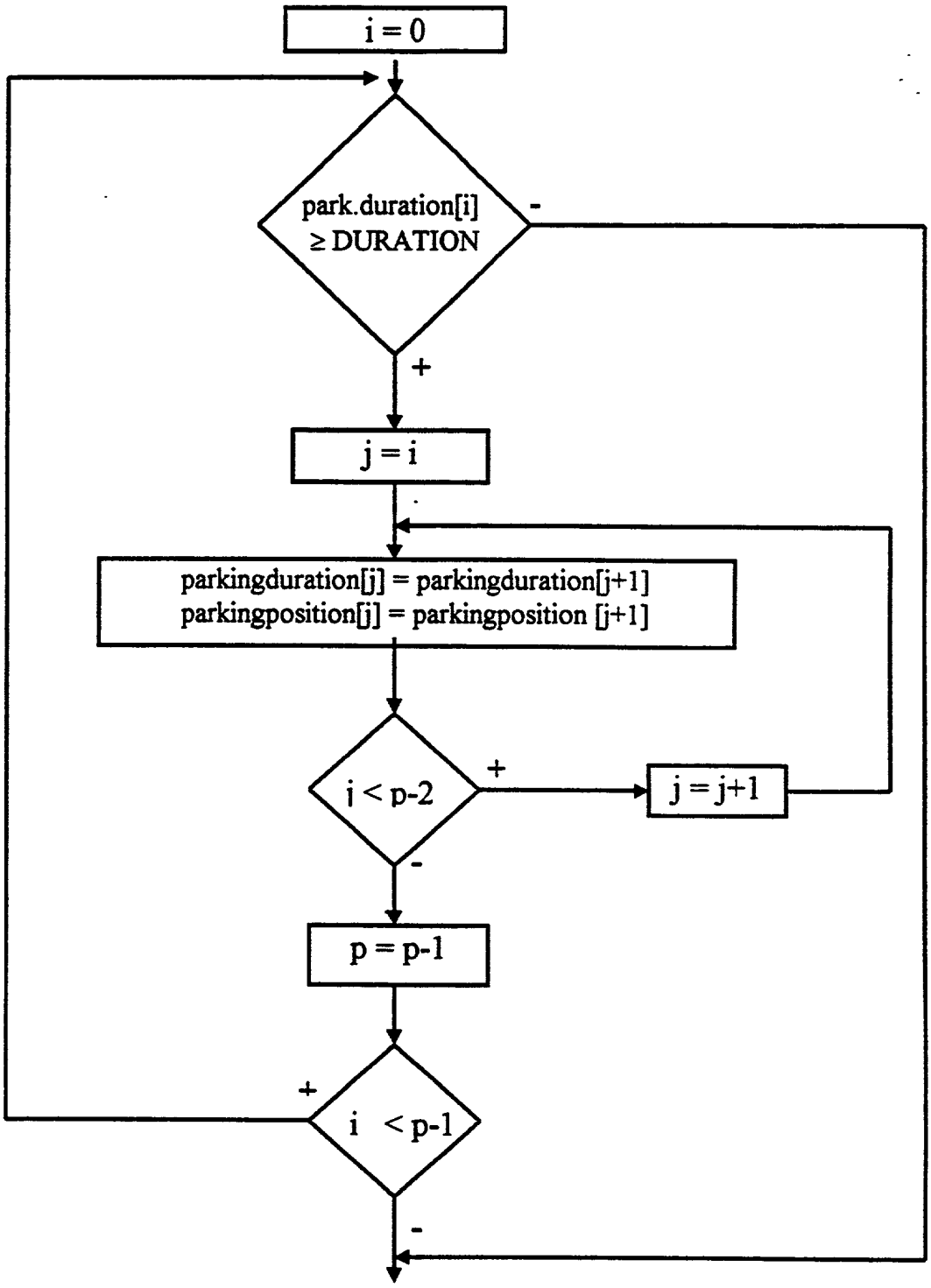

i = 0
park.duration[i]
≥ DURATION
-
+
j = i
parkingduration[j] = parkingduration[j+1]
parkingposition[j] = parkingposition [j+1]
i < p-2
+
j = j+1
-
p = p-1
+
i < p-1
-

Fig. 7B

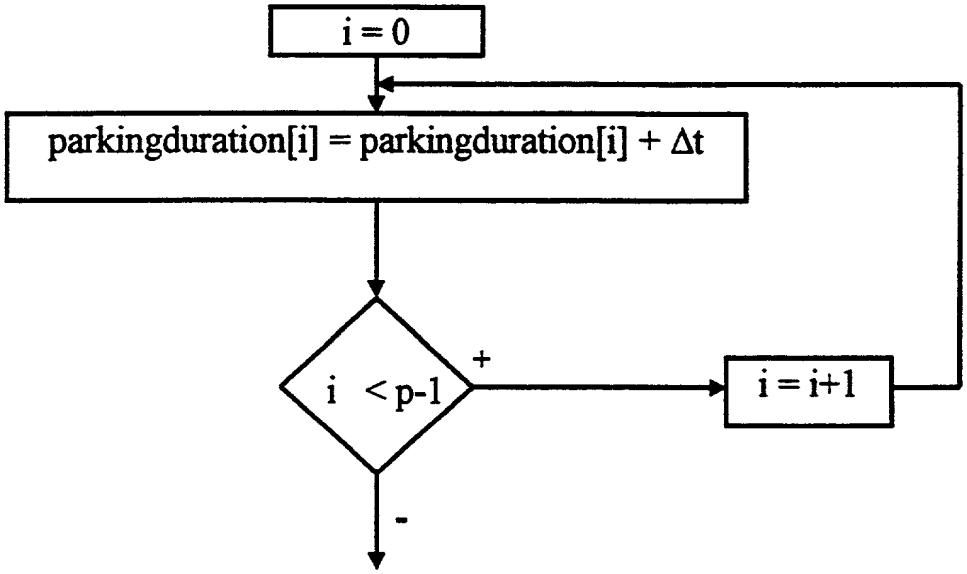


Fig. 7C

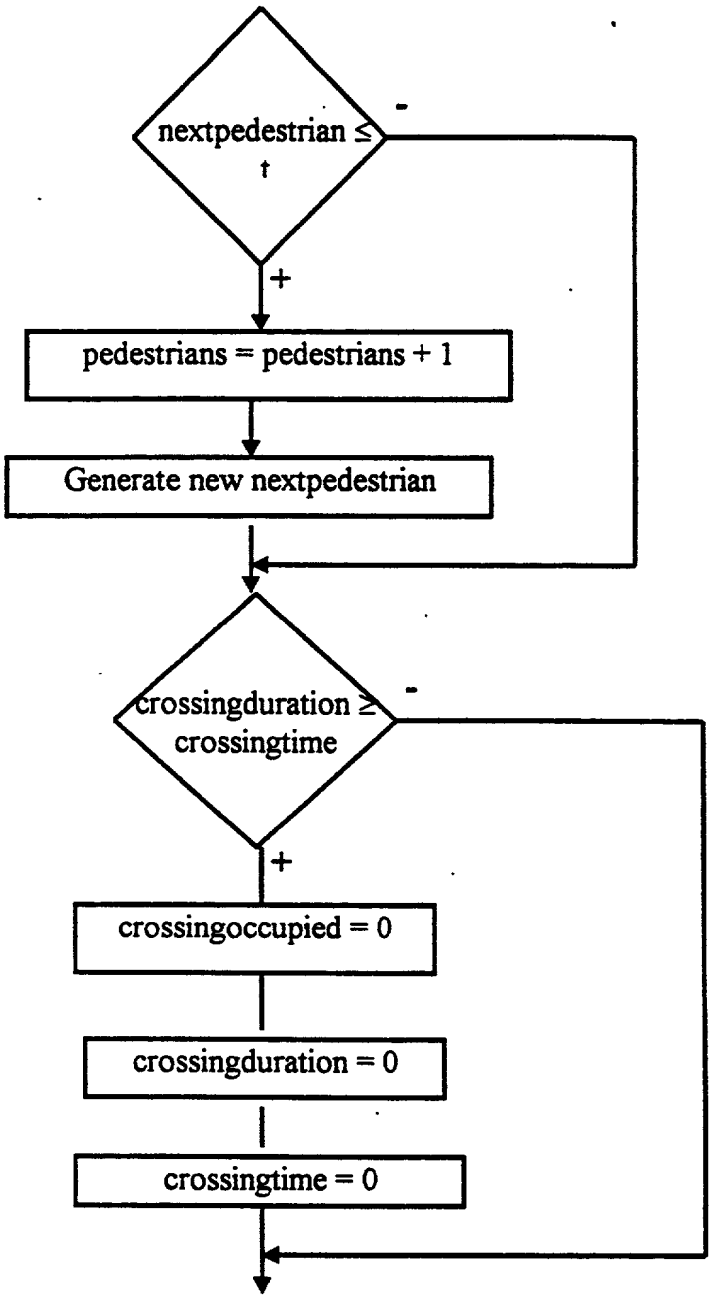

nextpedestrian ≤ t
-
+
pedestrians = pedestrians + 1
Generate new nextpedestrian
crossingduration ≥ crossingtime
-
+
crossingoccupied = 0
crossingduration = 0
crossingtime = 0

Fig. 8A

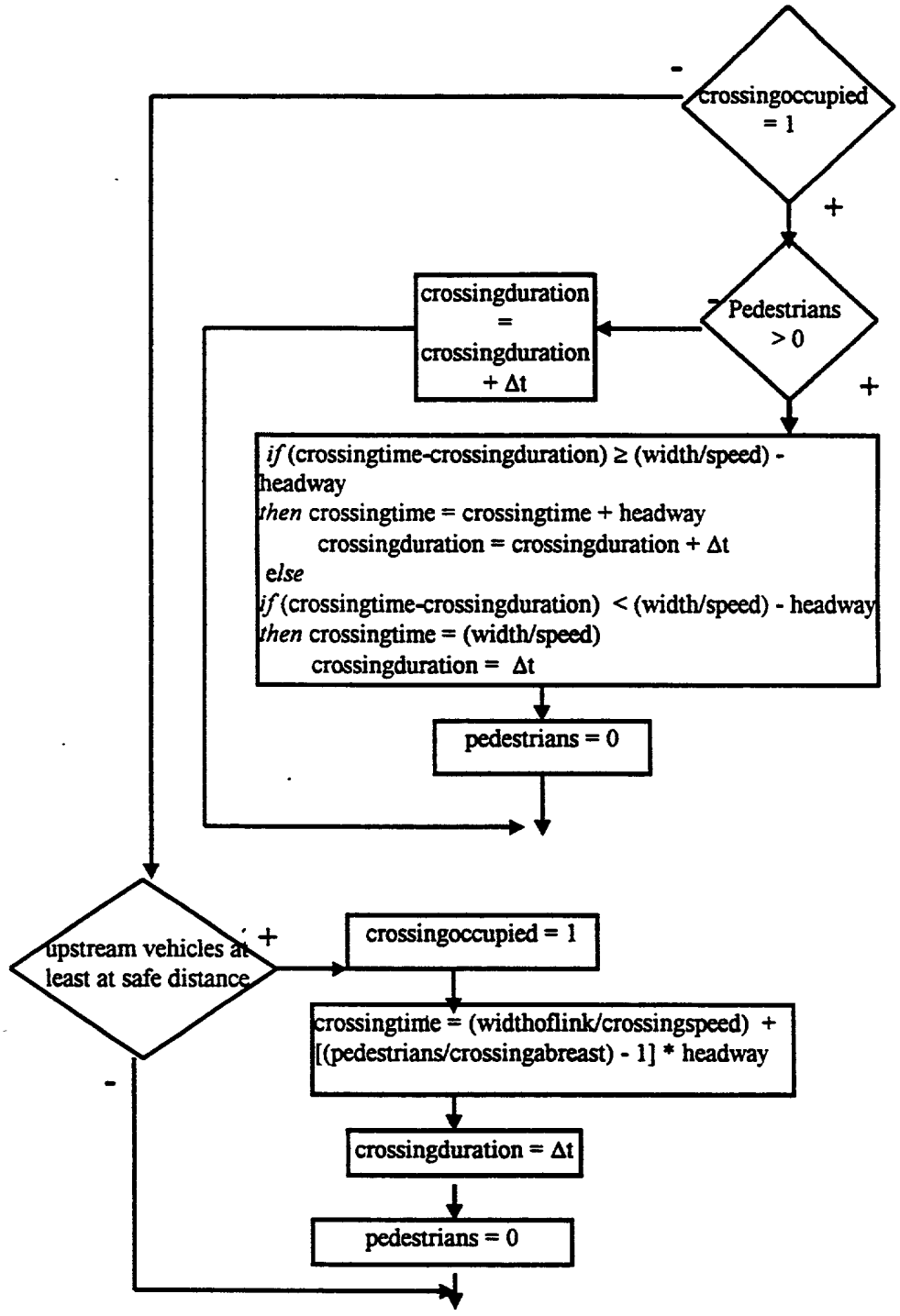

crossingoccupied = 1
-
+
Pedestrians > 0
-
+
crossingduration = crossingduration + Δt
if (crossingtime-crossingduration) ≥ (width/speed) - headway
then crossingtime = crossingtime + headway
crossingduration = crossingduration + Δt
else
if (crossingtime-crossingduration) < (width/speed) - headway
then crossingtime = (width/speed)
crossingduration = Δt
pedestrians = 0
upstream vehicles at least at safe distance
+
-
crossingoccupied = 1
crossingtime = (widthoflink/crossingspeed) + [(pedestrians/crossingabreast) - 1] * headway
crossingduration = Δt
pedestrians = 0

Fig. 8B

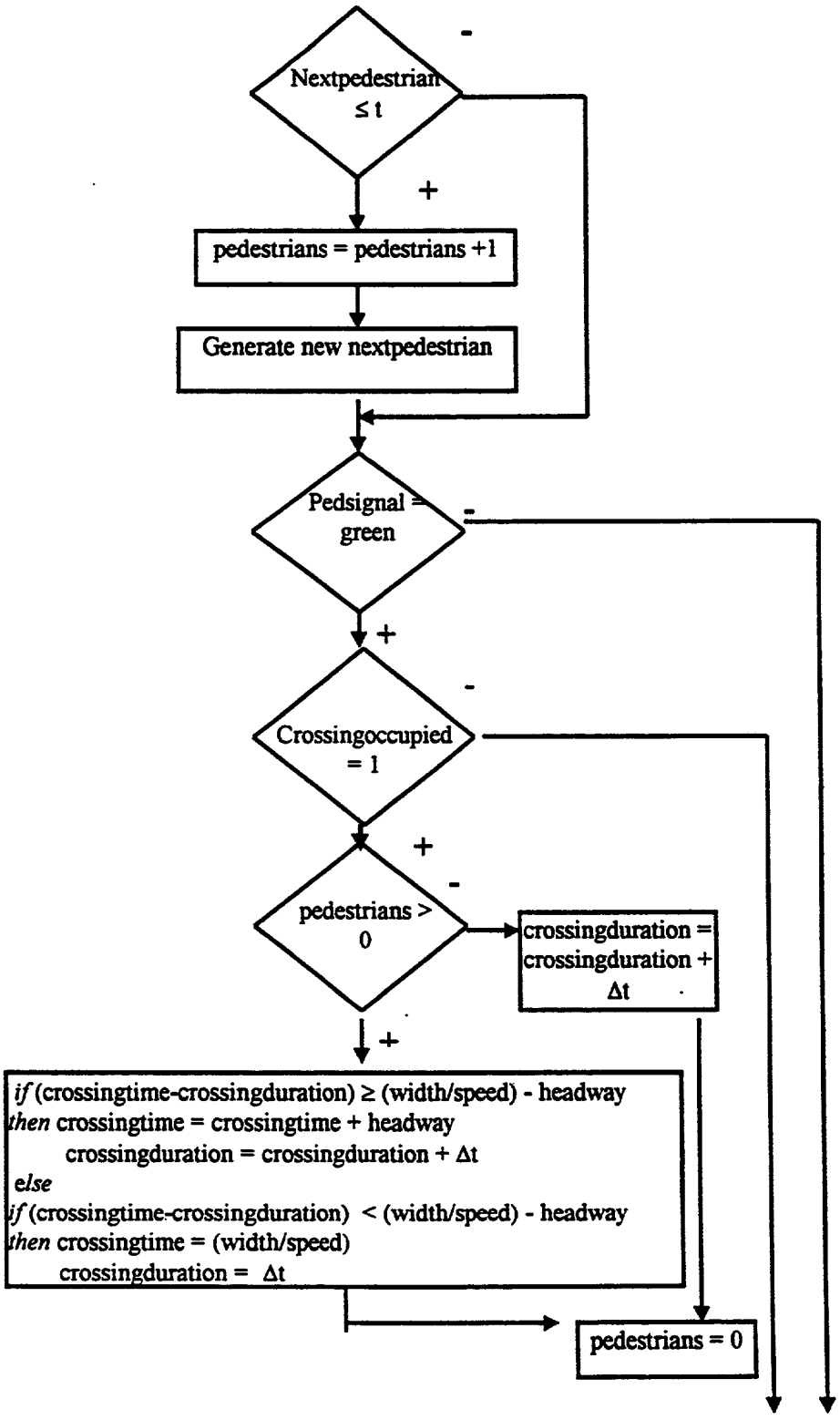

Nextpedestrian ≤ t
-
+
pedestrians = pedestrians +1
Generate new nextpedestrian
Pedsignal = green
-
+
Crossingoccupied = 1
-
+
pedestrians > 0
-
crossingduration = crossingduration + Δt
+
if (crossingtime-crossingduration) ≥ (width/speed) - headway
then crossingtime = crossingtime + headway
crossingduration = crossingduration + Δt
else
if (crossingtime-crossingduration) < (width/speed) - headway
then crossingtime = (width/speed)
crossingduration = Δt
pedestrians = 0

Fig. 8C

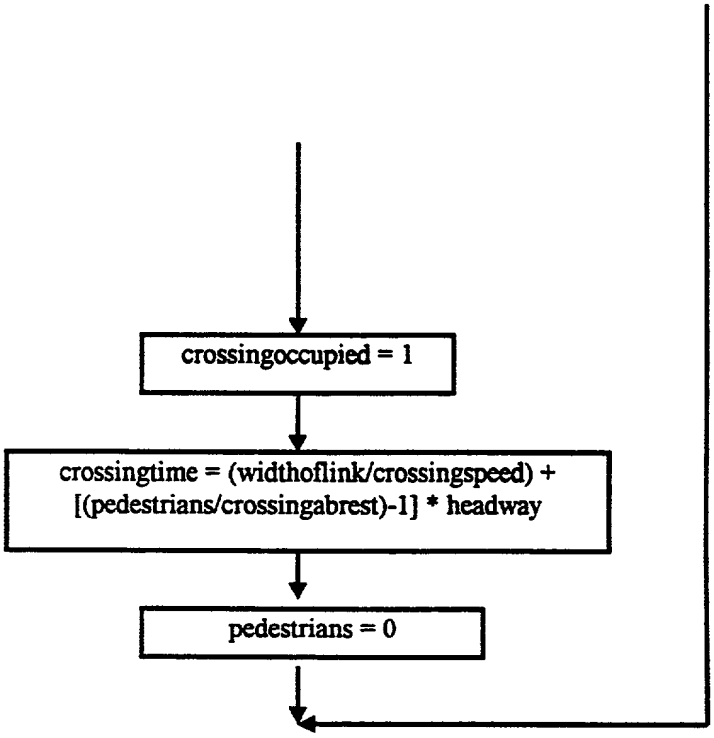

crossingoccupied = 1
crossingtime = (widthoflink/crossingspeed) +
[(pedestrians/crossingabrest)-1] * headway
pedestrians = 0

Fig. 8D

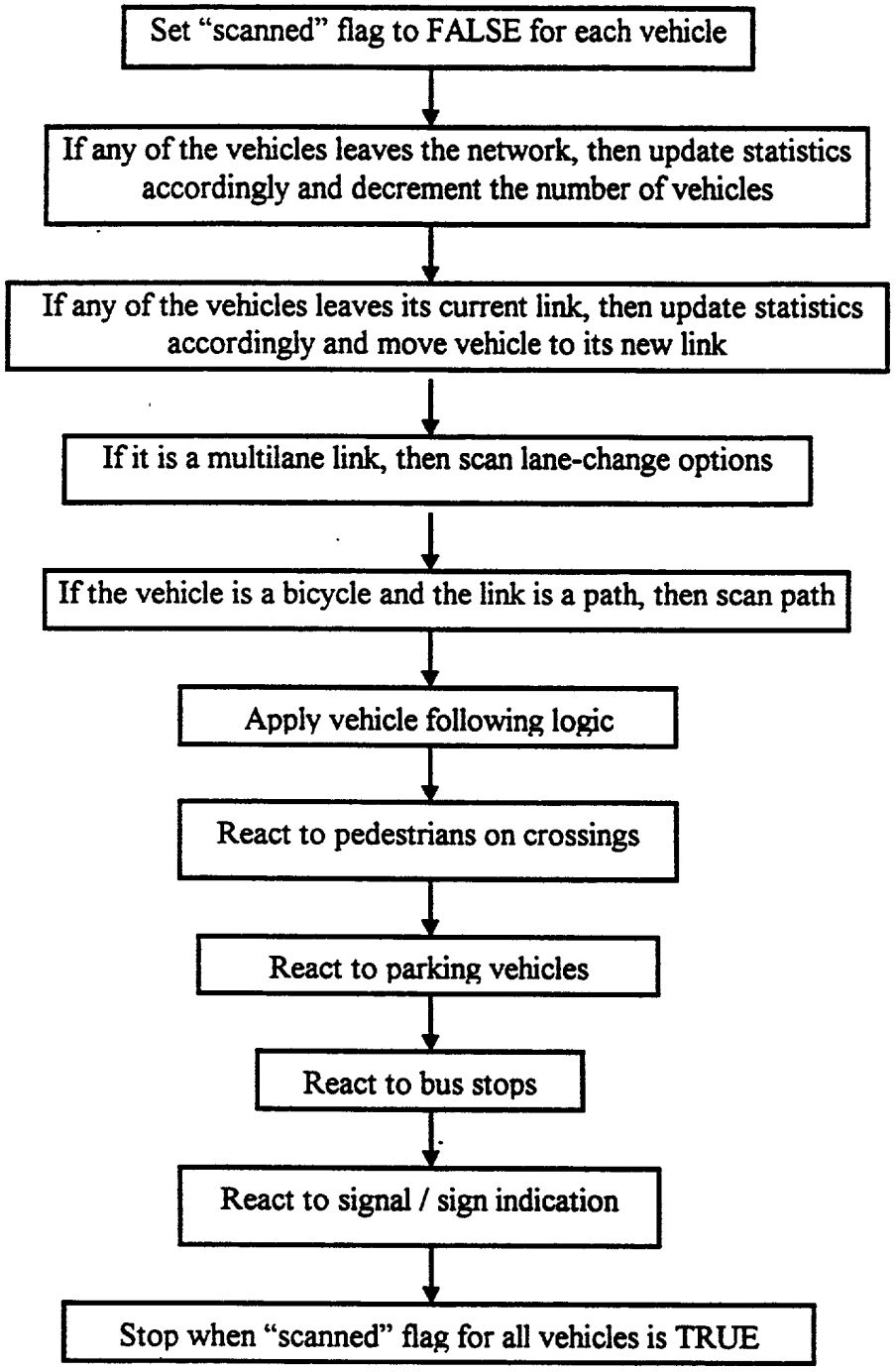

Set "scanned" flag to FALSE for each vehicle
If any of the vehicles leaves the network, then update statistics accordingly and decrement the number of vehicles
If any of the vehicles leaves its current link, then update statistics accordingly and move vehicle to its new link
If it is a multilane link, then scan lane-change options
If the vehicle is a bicycle and the link is a path, then scan path
Apply vehicle following logic
React to pedestrians on crossings
React to parking vehicles
React to bus stops
React to signal / sign indication
Stop when "scanned" flag for all vehicles is TRUE

Fig. 9

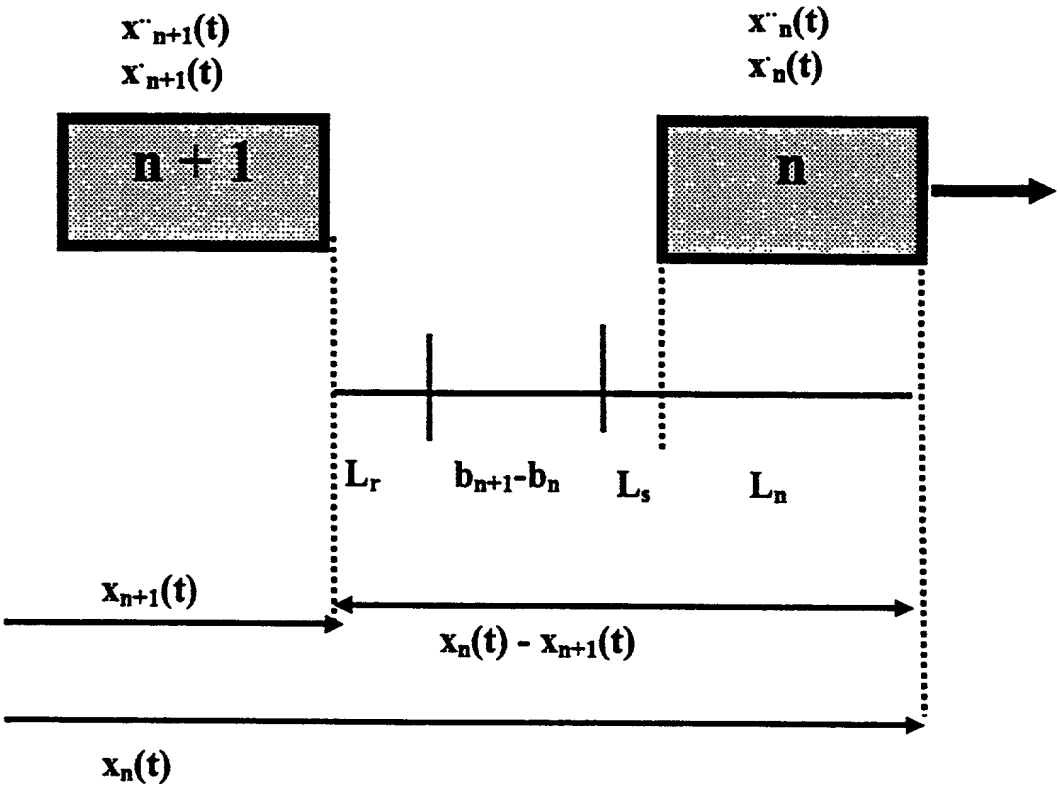

$\ddot{x}_{n+1}(t)$
$\dot{x}_{n+1}(t)$
$\ddot{x}_n(t)$
$\dot{x}_n(t)$
n + 1
n
L_r
$b_{n+1}-b_n$
L_s
L_n
$x_{n+1}(t)$
$x_n(t) - x_{n+1}(t)$
$x_n(t)$

Fig. 10

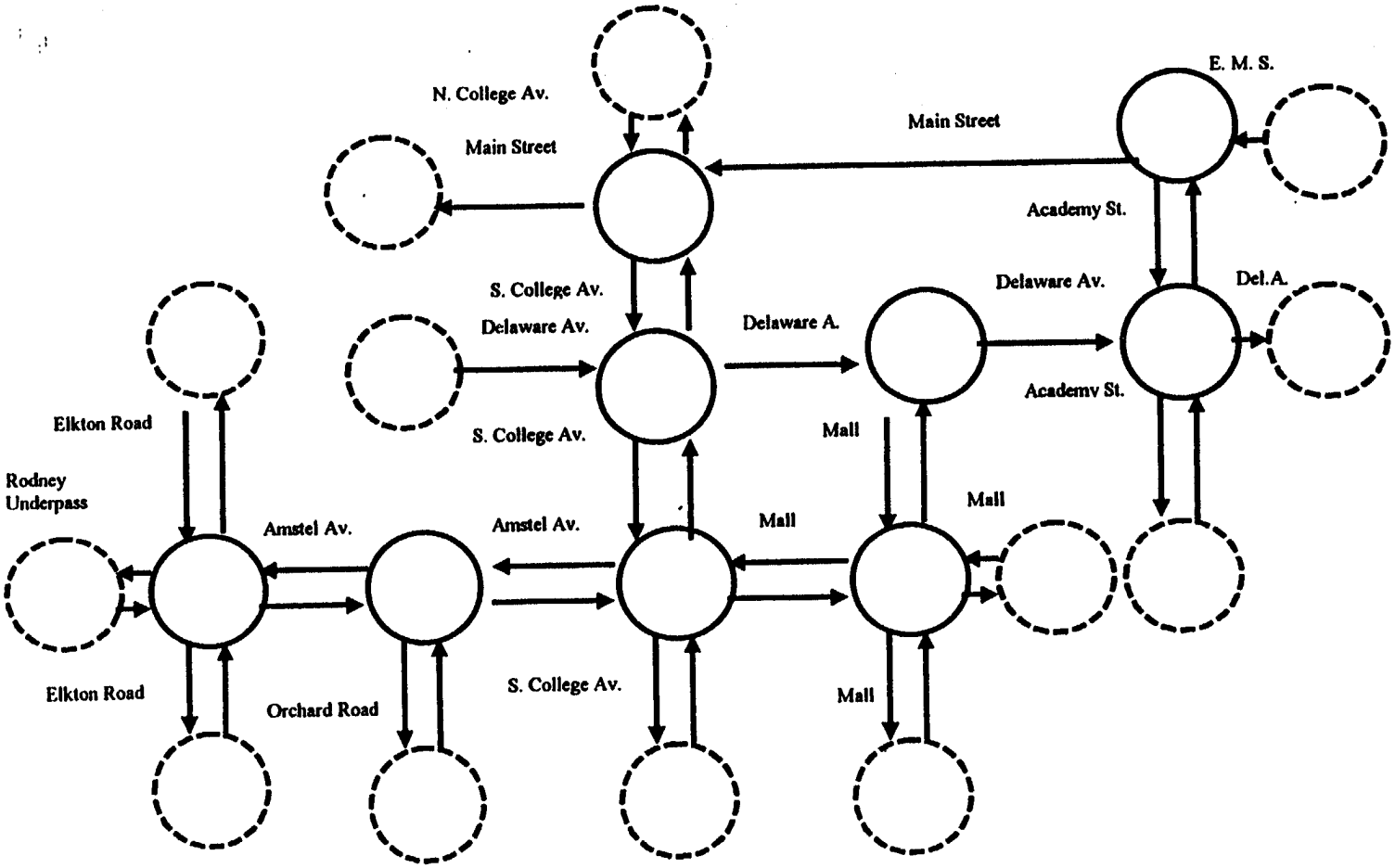


Fig. 11

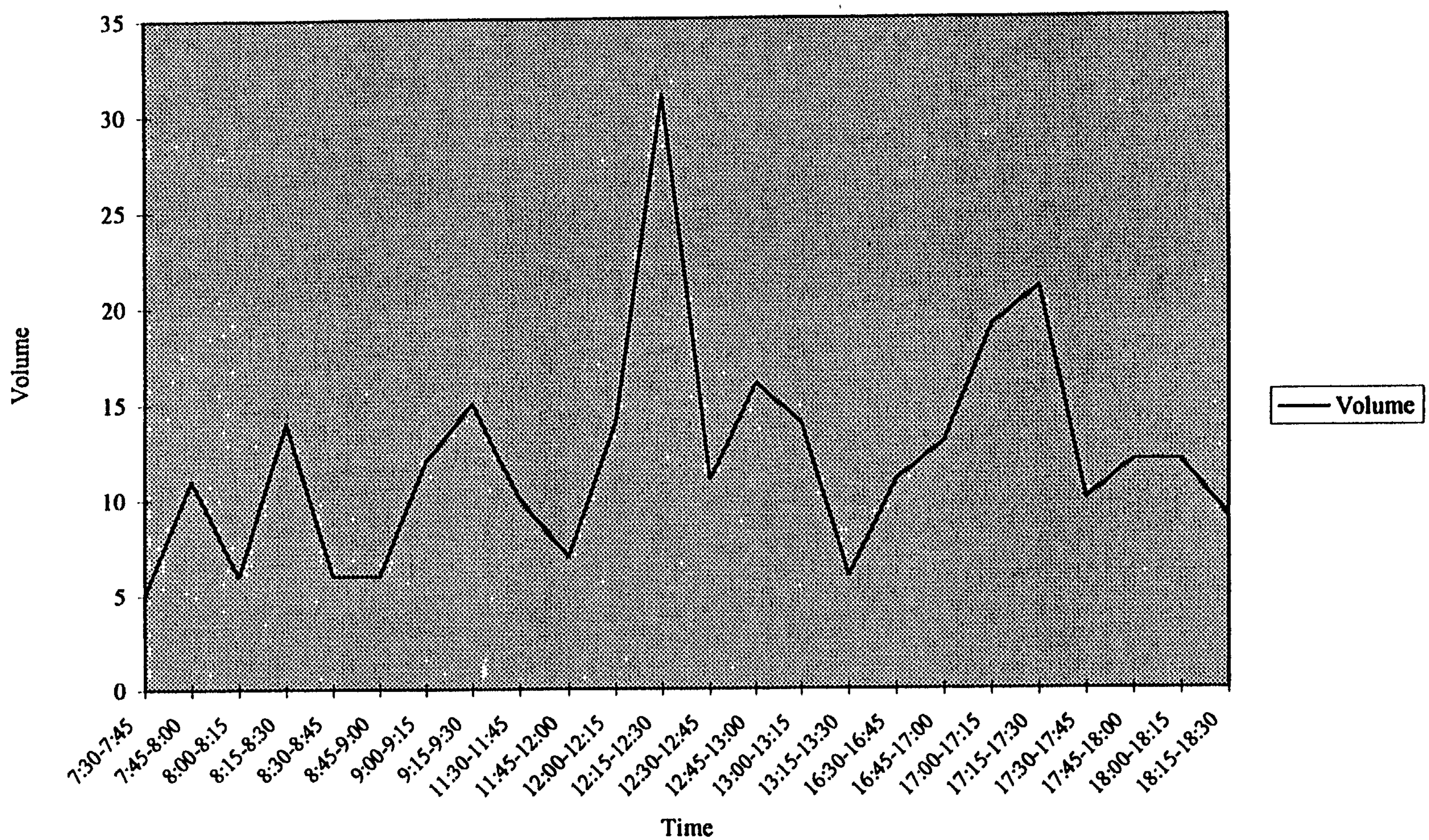

35
30
25
20
15
10
5
0
Volume
7:30-7:45
7:45-8:00
8:00-8:15
8:15-8:30
8:30-8:45
8:45-9:00
9:00-9:15
9:15-9:30
11:30-11:45
11:45-12:00
12:00-12:15
12:15-12:30
12:30-12:45
12:45-13:00
13:00-13:15
13:15-13:30
16:30-16:45
16:45-17:00
17:00-17:15
17:15-17:30
17:30-17:45
17:45-18:00
18:00-18:15
18:15-18:30
Time
Volume

Fig. 12

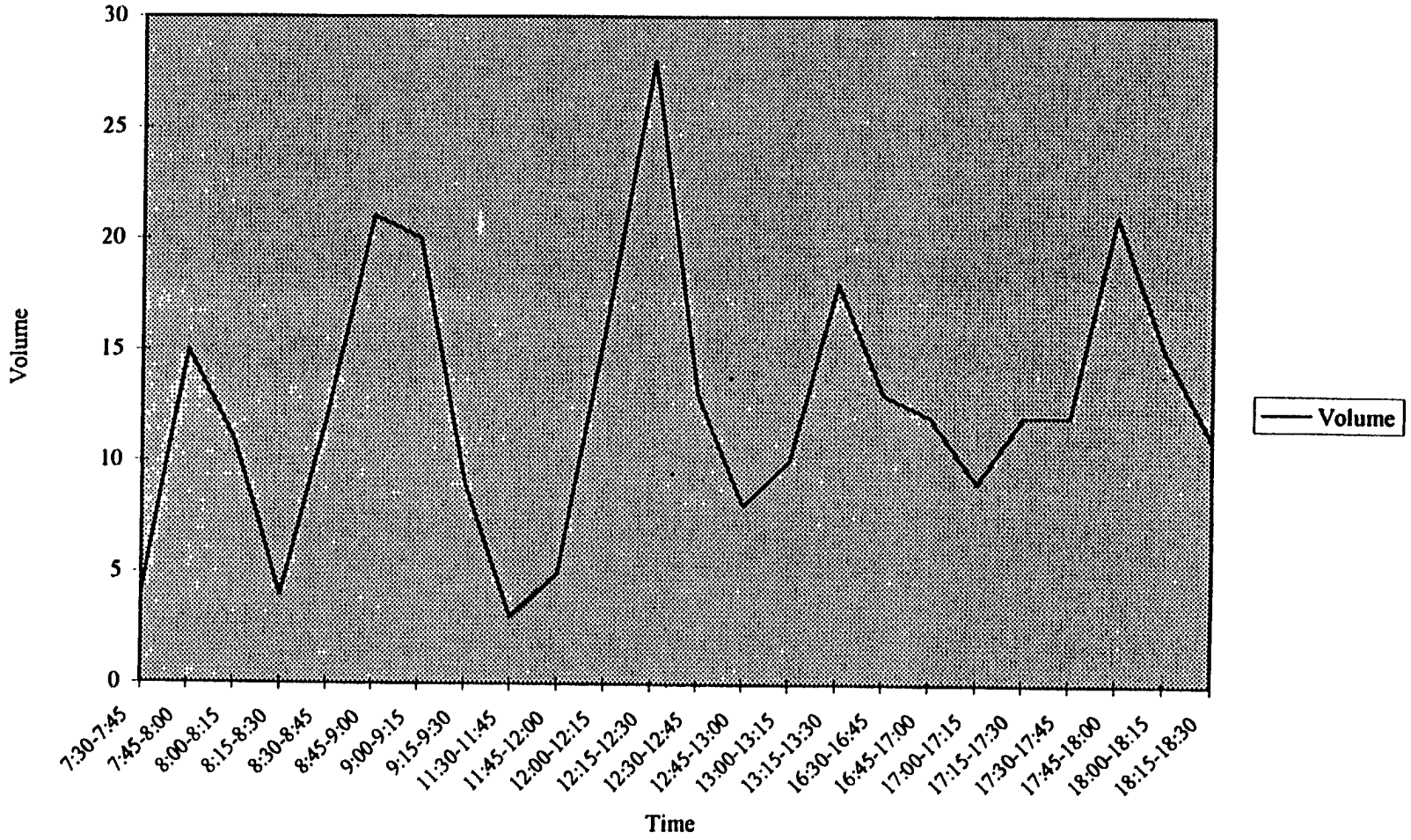


Fig. 13

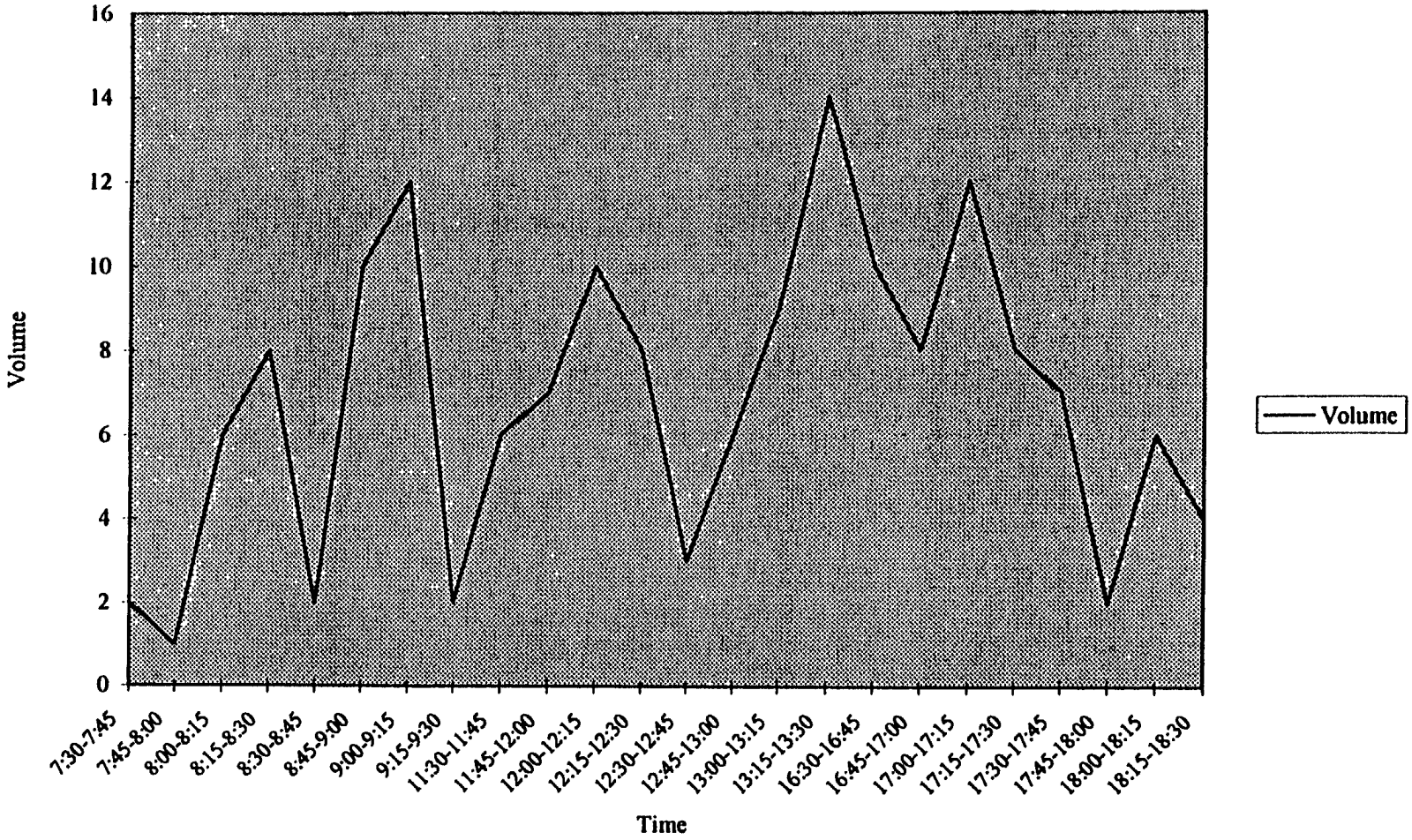


Fig. 14

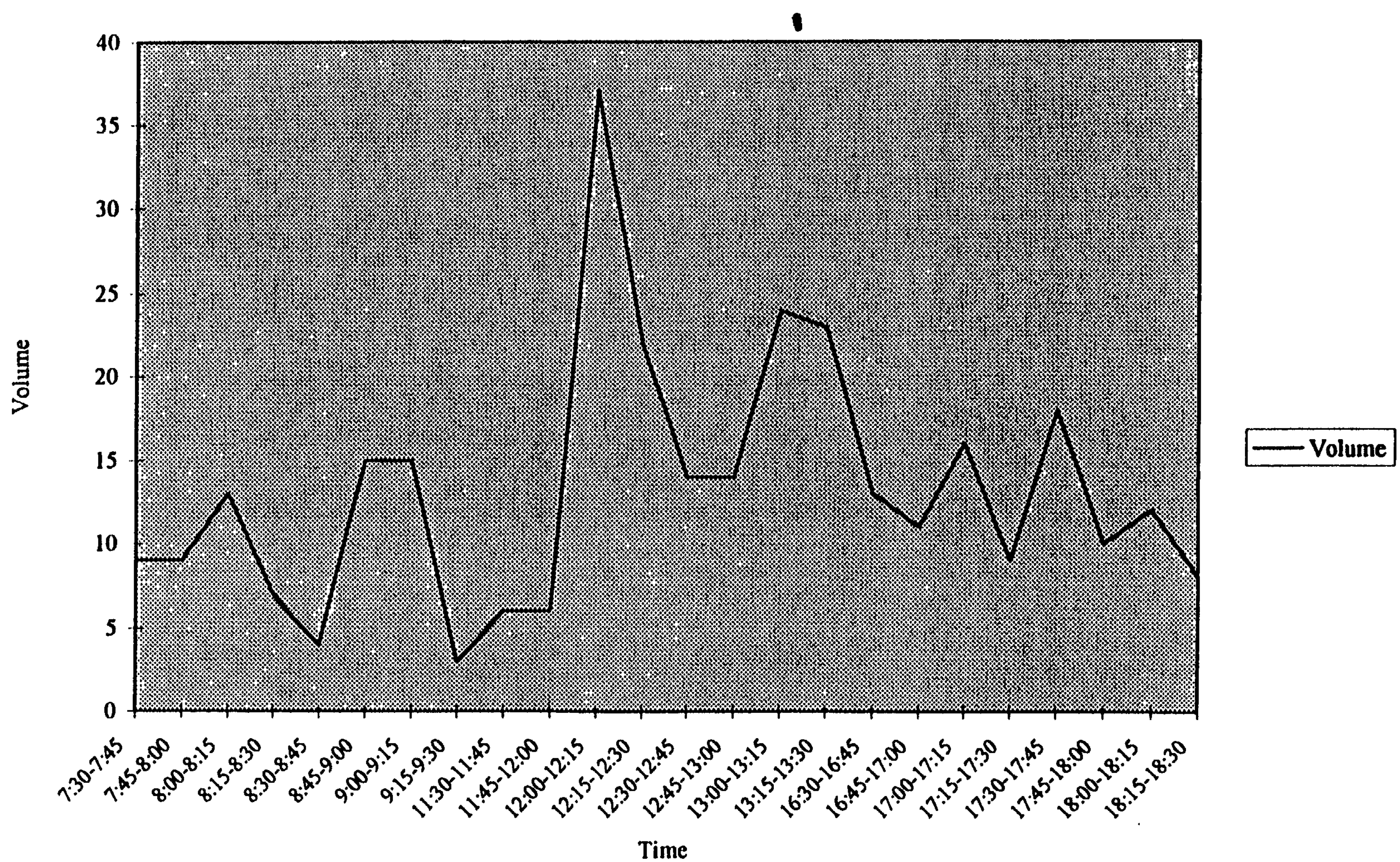

40
35
30
25
20
15
10
5
0
Volume
Volume
7:30-7:45
7:45-8:00
8:00-8:15
8:15-8:30
8:30-8:45
8:45-9:00
9:00-9:15
9:15-9:30
11:30-11:45
11:45-12:00
12:00-12:15
12:15-12:30
12:30-12:45
12:45-13:00
13:00-13:15
13:15-13:30
16:30-16:45
16:45-17:00
17:00-17:15
17:15-17:30
17:30-17:45
17:45-18:00
18:00-18:15
18:15-18:30
Time

Fig. 15

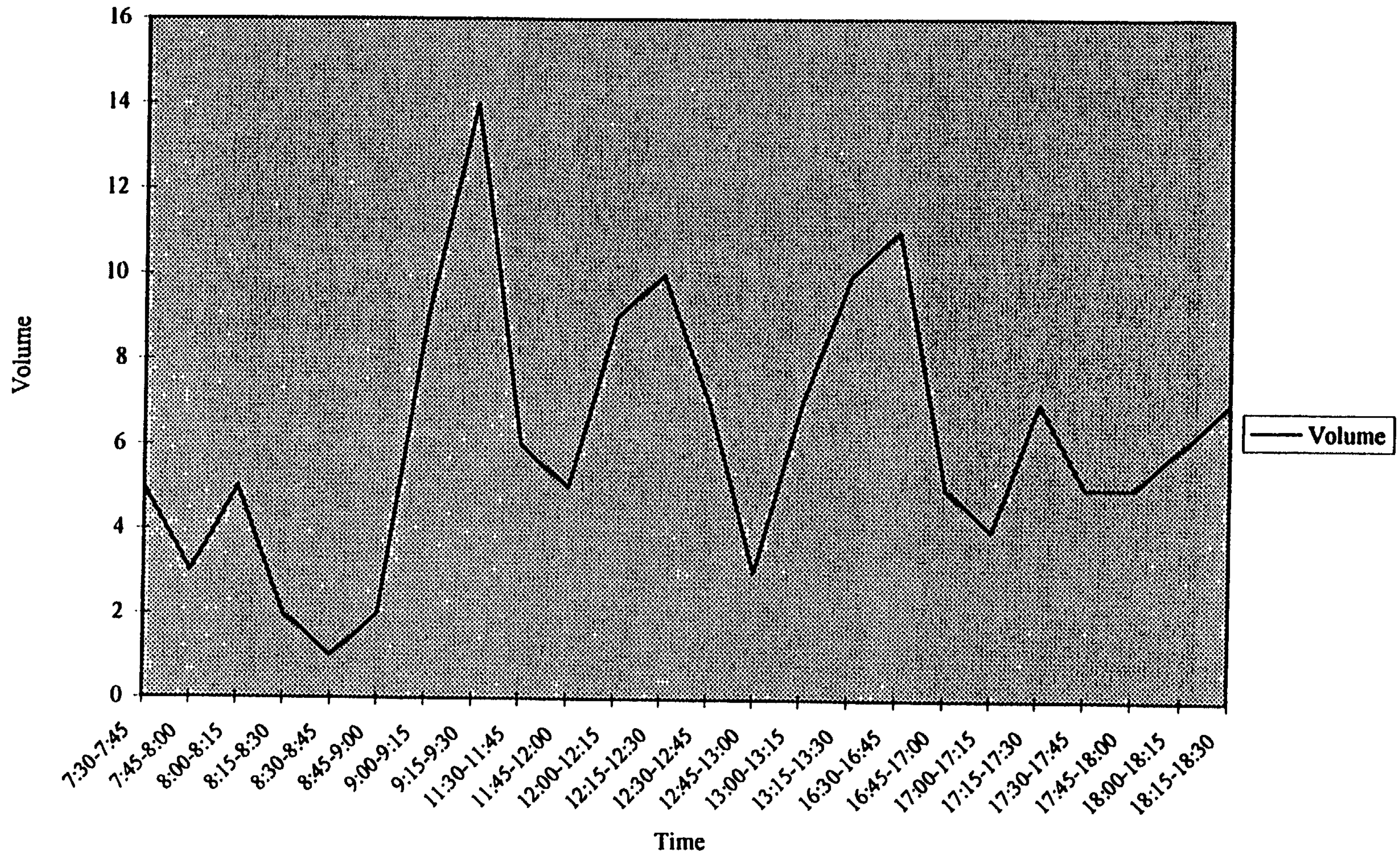

16
14
12
10
8
6
4
2
0
Volume
7:30-7:45
7:45-8:00
8:00-8:15
8:15-8:30
8:30-8:45
8:45-9:00
9:00-9:15
9:15-9:30
11:30-11:45
11:45-12:00
12:00-12:15
12:15-12:30
12:30-12:45
12:45-13:00
13:00-13:15
13:15-13:30
16:30-16:45
16:45-17:00
17:00-17:15
17:15-17:30
17:30-17:45
17:45-18:00
18:00-18:15
18:15-18:30
Time
Volume

Fig. 16

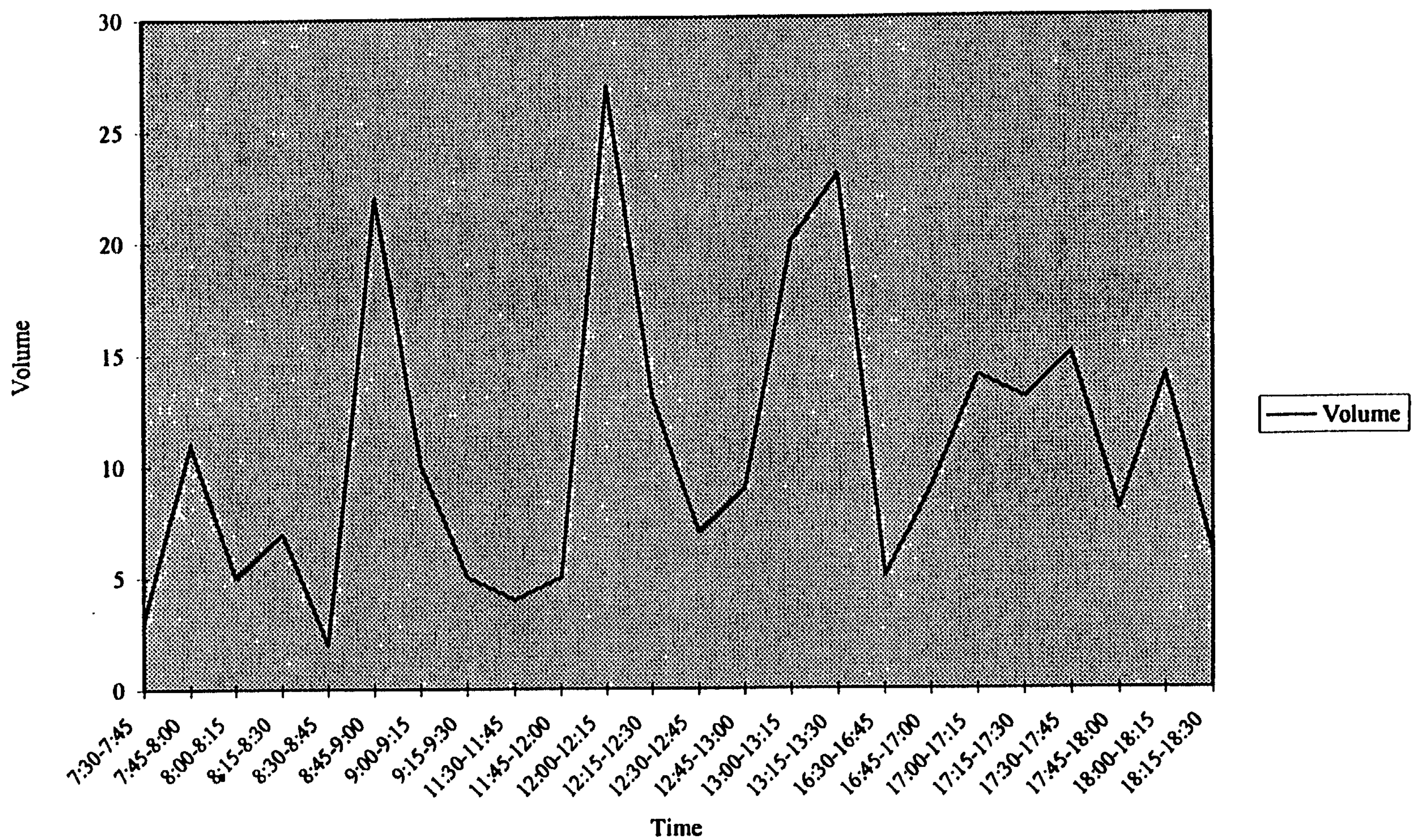

Volume
30
25
20
15
10
5
0
7:30-7:45
7:45-8:00
8:00-8:15
8:15-8:30
8:30-8:45
8:45-9:00
9:00-9:15
9:15-9:30
11:30-11:45
11:45-12:00
12:00-12:15
12:15-12:30
12:30-12:45
12:45-13:00
13:00-13:15
13:15-13:30
16:30-16:45
16:45-17:00
17:00-17:15
17:15-17:30
17:30-17:45
17:45-18:00
18:00-18:15
18:15-18:30
Time
Volume

Fig. 17

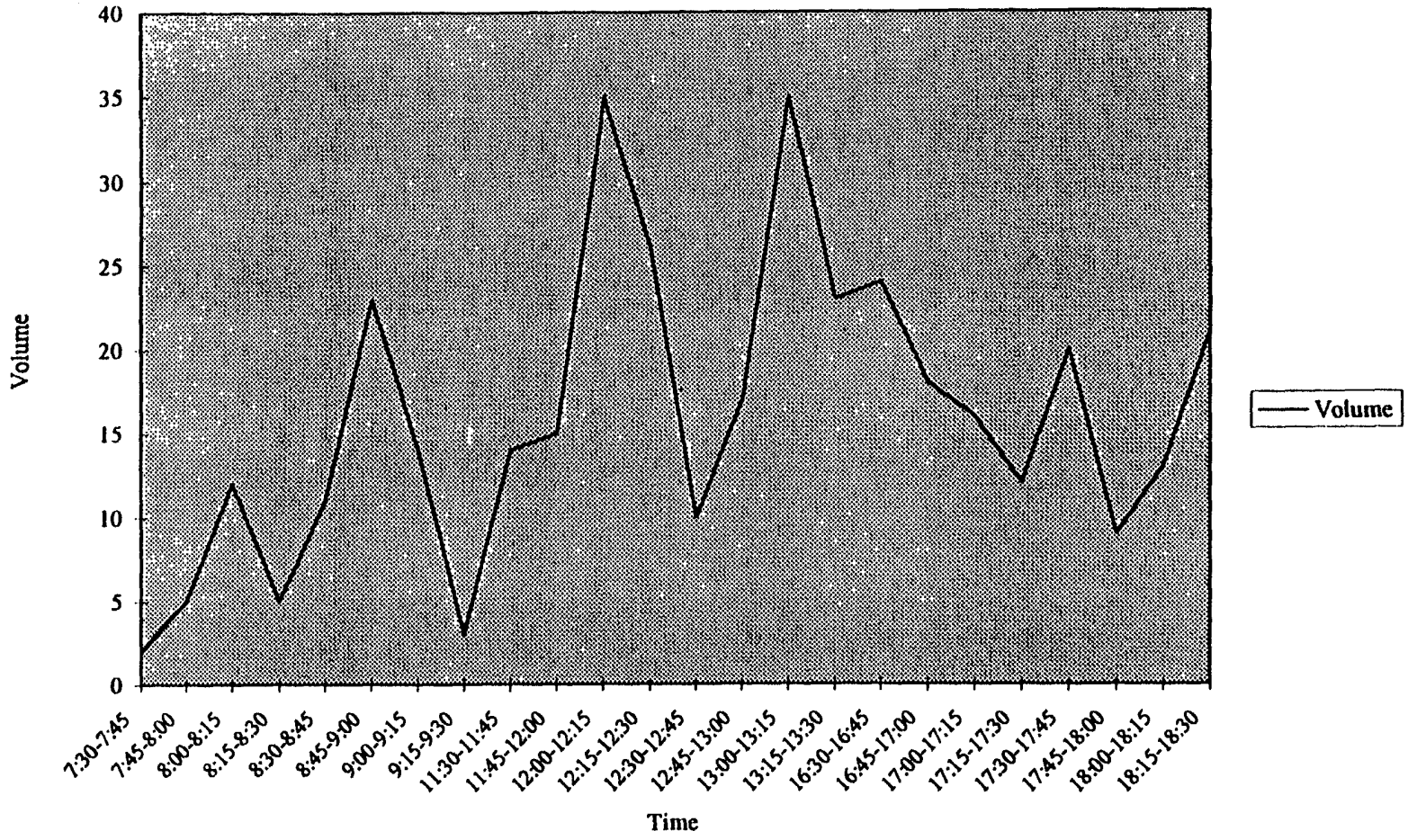


Fig. 18

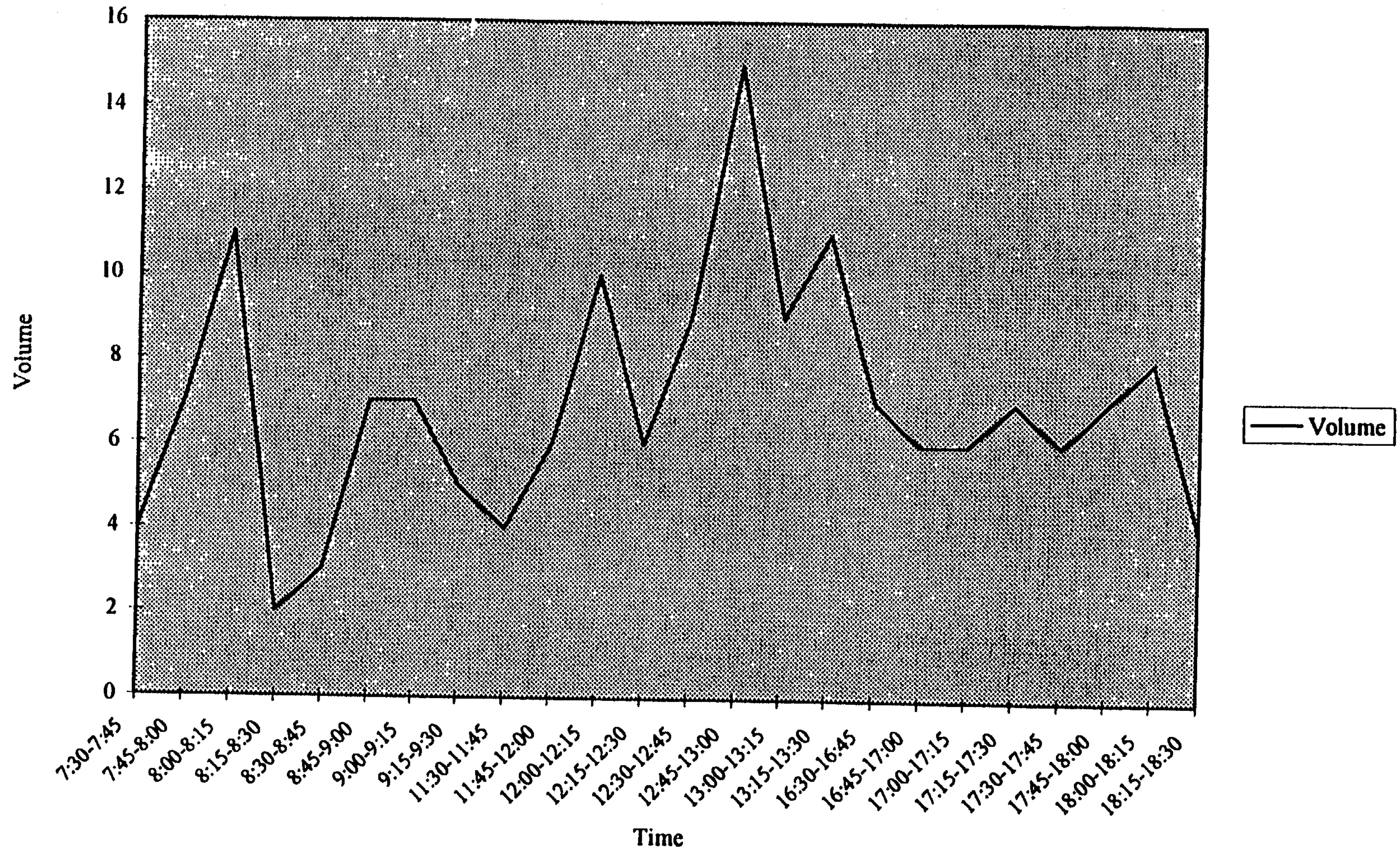
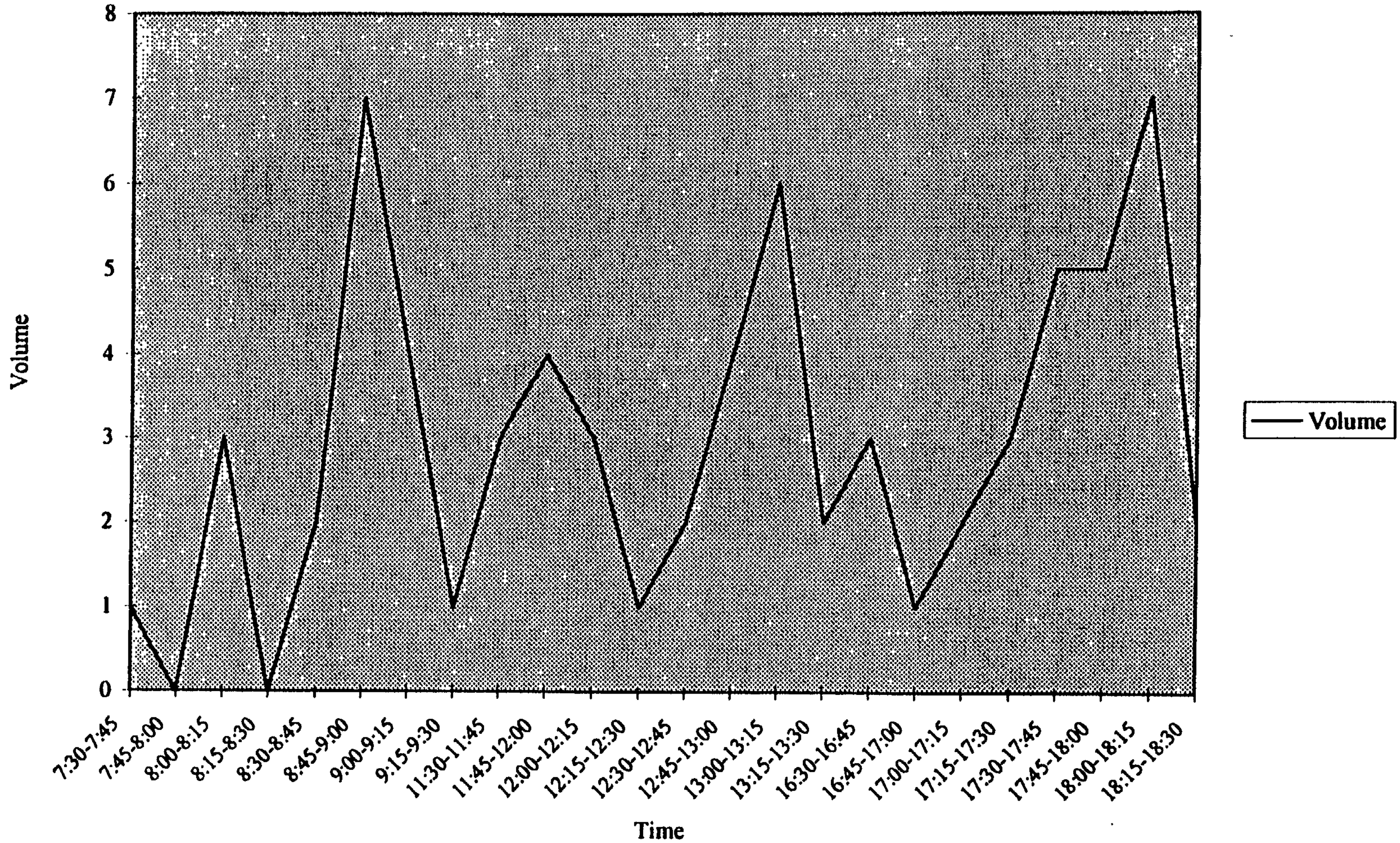

Volume
16
14
12
10
8
6
4
2
0
7:30-7:45
7:45-8:00
8:00-8:15
8:15-8:30
8:30-8:45
8:45-9:00
9:00-9:15
9:15-9:30
11:30-11:45
11:45-12:00
12:00-12:15
12:15-12:30
12:30-12:45
12:45-13:00
13:00-13:15
13:15-13:30
16:30-16:45
16:45-17:00
17:00-17:15
17:15-17:30
17:30-17:45
17:45-18:00
18:00-18:15
18:15-18:30
Time
Volume

Fig. 19



Volume
8
7
6
5
4
3
2
1
0
7:30-7:45
7:45-8:00
8:00-8:15
8:15-8:30
8:30-8:45
8:45-9:00
9:00-9:15
9:15-9:30
11:30-11:45
11:45-12:00
12:00-12:15
12:15-12:30
12:30-12:45
12:45-13:00
13:00-13:15
13:15-13:30
16:30-16:45
16:45-17:00
17:00-17:15
17:15-17:30
17:30-17:45
17:45-18:00
18:00-18:15
18:15-18:30
Time
Volume

Fig. 20

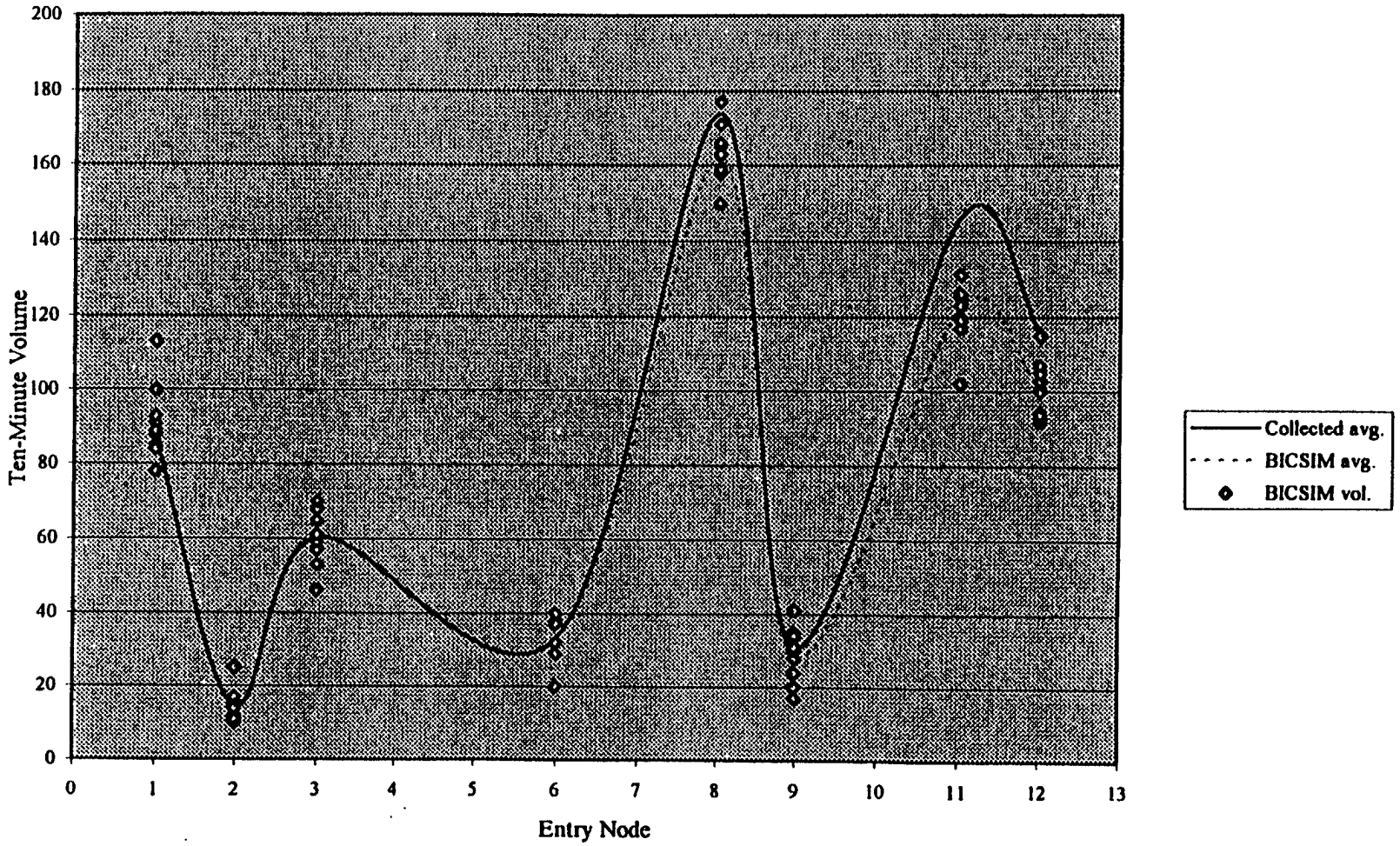


Fig. 21

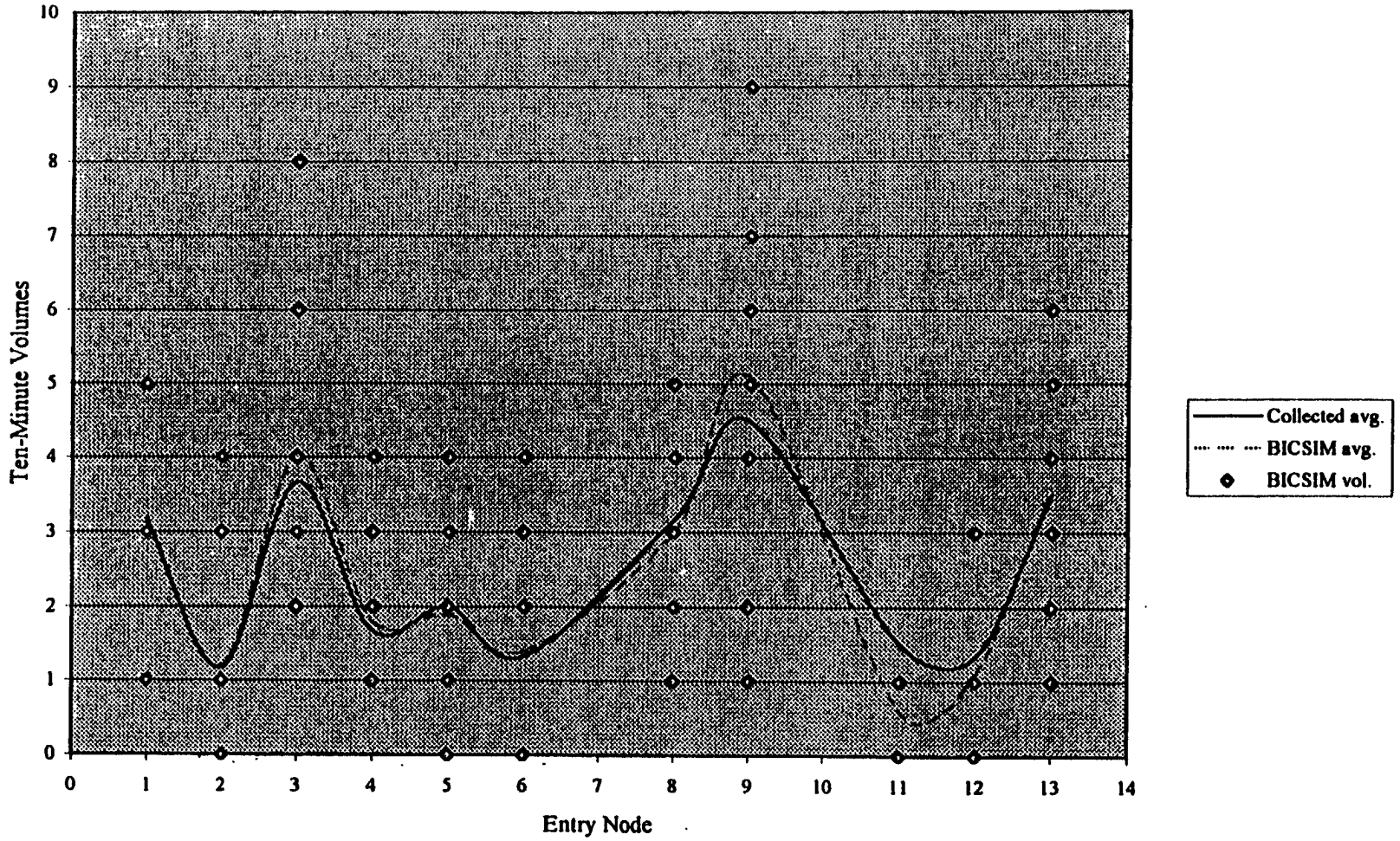


Fig. 22

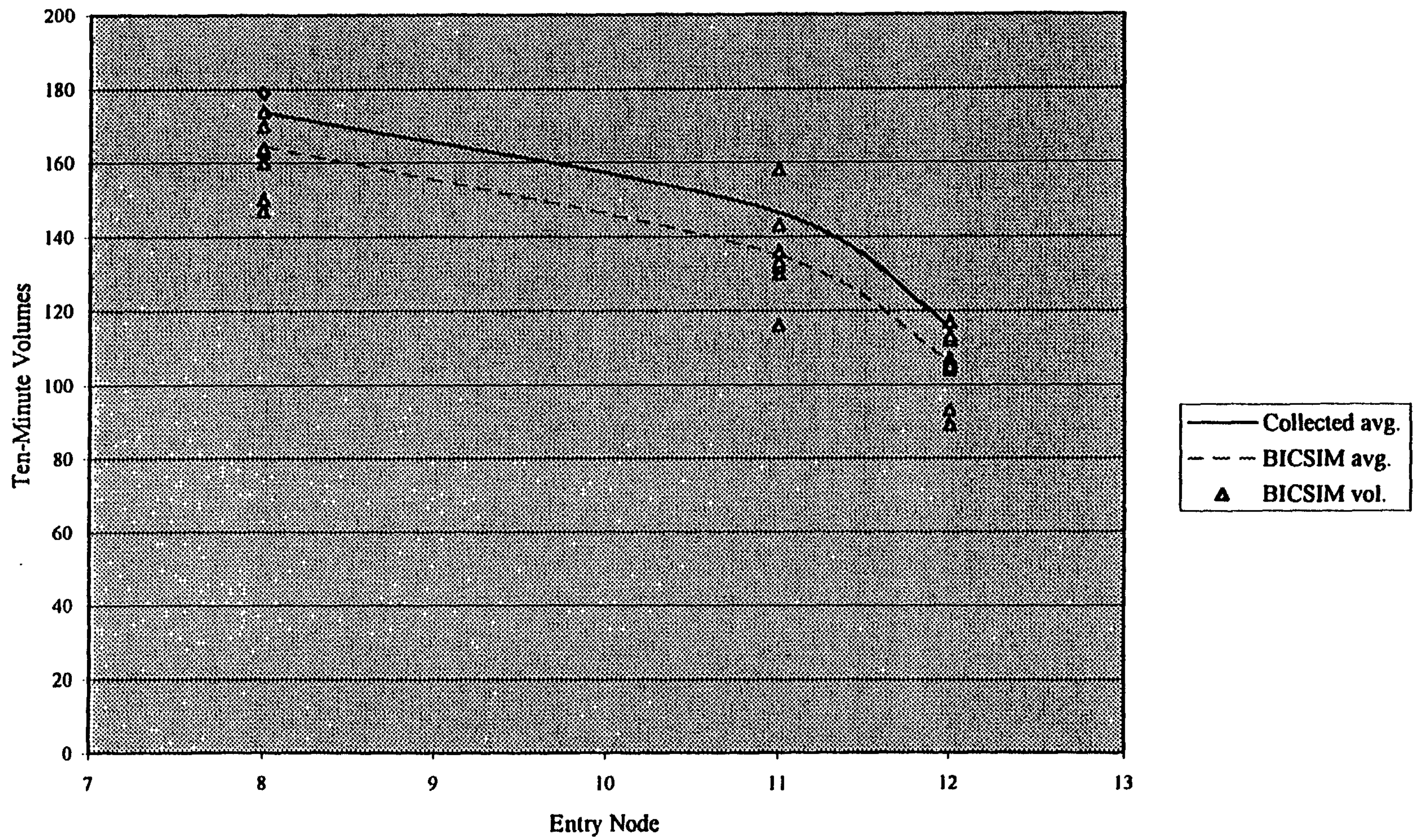

Ten-Minute Volumes
200
180
160
140
120
100
80
60
40
20
0
7
8
9
10
11
12
13
Entry Node
Collected avg.
BICSIM avg.
BICSIM vol.

Fig. 23

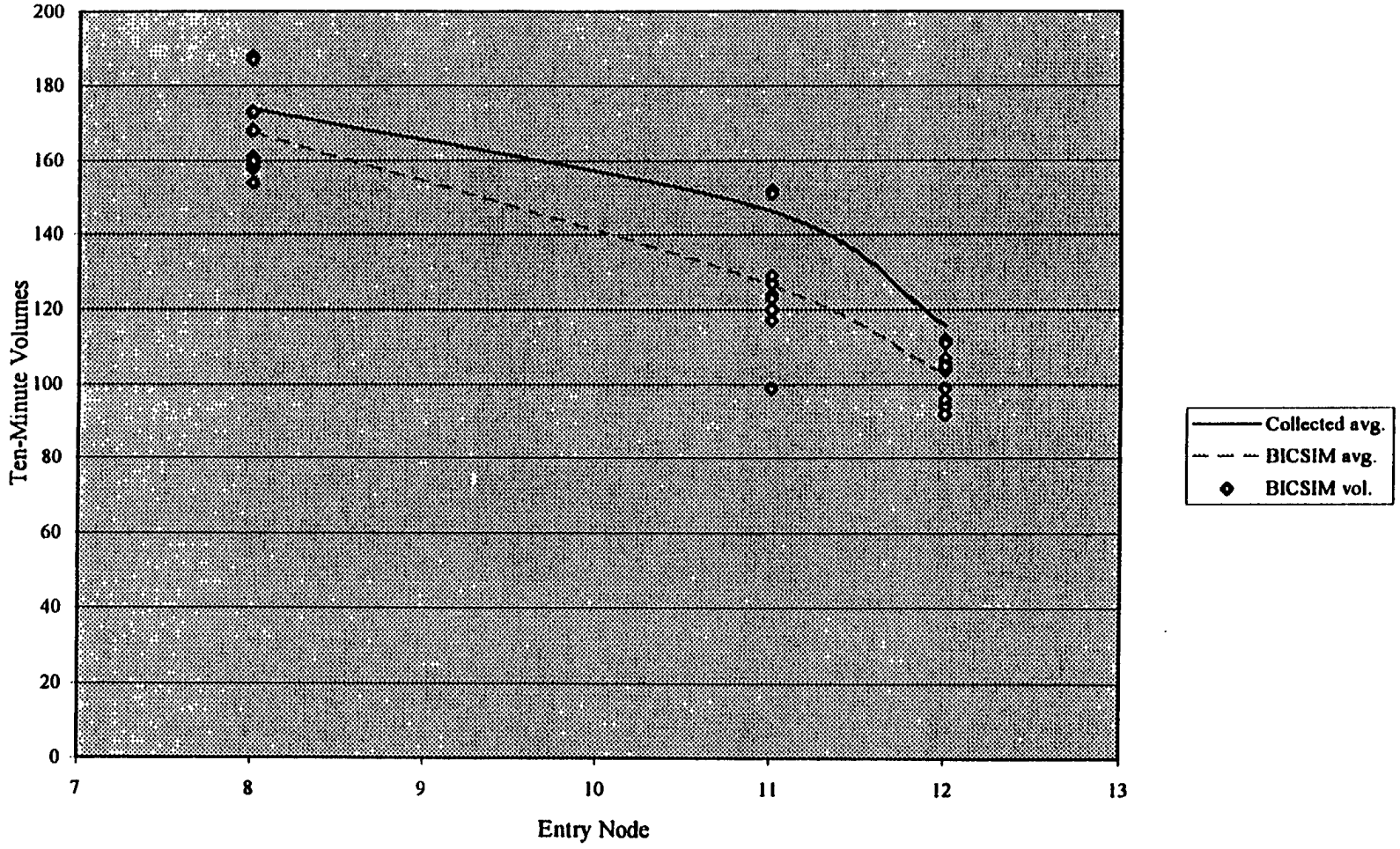
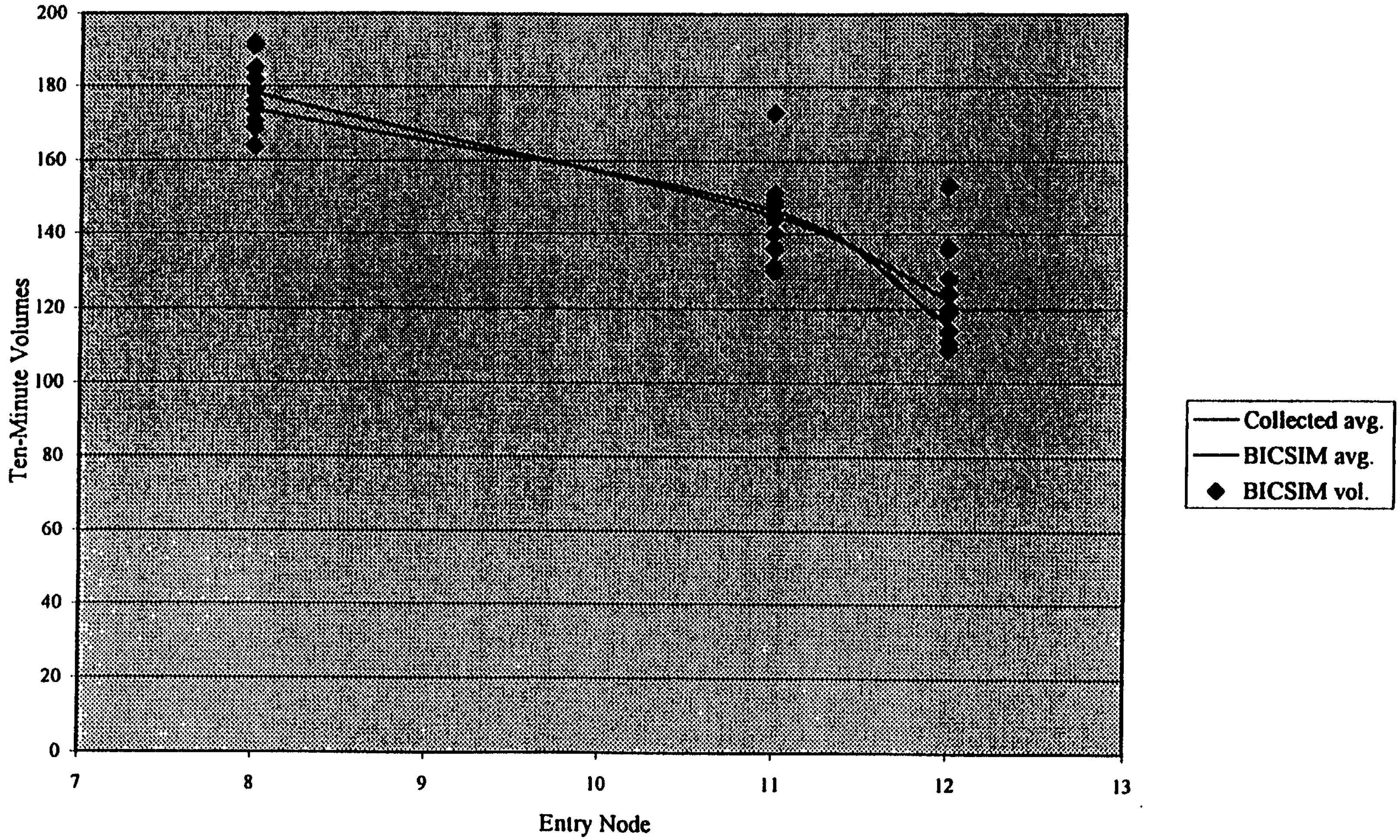

200
180
160
140
120
100
80
60
40
20
0
Ten-Minute Volumes
7
8
9
10
11
12
13
Entry Node
Collected avg.
BICSIM avg.
BICSIM vol.

Fig. 24



Ten-Minute Volumes
200
180
160
140
120
100
80
60
40
20
0
7
8
9
10
11
12
13
Entry Node
Collected avg.
BICSIM avg.
BICSIM vol.

Fig. 25

TABLE 1

		Major Road						Minor Road					
		Right		Left		Thru		Right		Left		Thru	
		S	M	S	M	S	M	S	M	S	M	S	M
In Bike Coming From	Adjacent	2.5	2	x	x	x	x	3	2	x	x	x	x
	Left	x	x	x	x	x	x	4	3	4.5	4	4	3.5
	Right	x	x	x	x	x	x	x	x	4.5	4	4	3.5
	Oppose	x	x	4.5	4	x	x	x	x	4.5	4	x	x
In MV Coming From	Left	x	x	x	x	x	x	6	5	6.5	6	6	5.5
	Right	x	x	x	x	x	x	x	x	6.5	6	6	5.5
	Oppose	x	x	6.5	6	x	x	x	x	6.5	6	x	x

S=stopped *M=moving*

TABLE 2

		Major Road						Minor Road					
		Right		Left		Thru		Right		Left		Thru	
		S	M	S	M	S	M	S	M	S	M	S	M
In Bike Coming From	Left	x	x	x	x	x	x	5	3	6	4	5.5	3.5
	Right	x	x	x	x	x	x	x	x	6	4	5.5	3.5
	Oppose	x	x	6	4	x	x	x	x	6	4	x	x
In MV Coming From	Left	x	x	x	x	x	x	7	5	8	6	7.5	5.5
	Right	x	x	x	x	x	x	x	x	8	6	7.5	5.5
	Oppose	x	x	8	6	x	x	x	x	8	6	x	x

S=stopped *M=moving*

TABLE 3

	Parking	Leaving
1	3.72	2.38
2	3.99	2.33
3	3.17	2.74
4	2.96	5.07
5	4.88	3.26
6	3.76	6.21
7	2.91	8.51
8	2.59	3.42
9	6.95	3.65
10	4.58	3.72
11	2.94	3.94
12	6.18	5.51
13	8.94	3.6
14	4.05	6.37
15	2.69	5.38
Average:	4.287333	4.406
Average for both:		4.346667

Collected on Main Street, Newark, DE , July 10, 1997

TABLE 4

Node #	Buslines	Name	Headw.	Name	Headw.	Name	Headw.	Name	Headw.	Name	Headw.	Name	Headw
1	3	33	1800	UD(T/T)	1200	UD(E1)	2400						
2													
3	2	UD(P/S)	2400	UD(S)	600/1200**								
4													
5													
6	5	302	12000	6	1800	UD(L)	780/1500*	UD(E1)	2400	UD(E2)	2100		
7													
8	4	33	1800	34	3000	6	1800	UD(E2)	2100				
9	1	UD(P/S)	2400										
10													
11	6	33	1800	UD(T/T)	1200	UD(P/S)	2400	UD(L)	780/1500*	UD(E1)	2400	UD(E2)	2100
12	2	33	1800	6	1800								
13													

Notes: *(1) Headways are in seconds.*
(2) Headways for non-UD buses are approximate.

* *changes at 2:15 pm*

** *changes at 2:04 pm*